ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/339977379

Predicting Citywide Road Traffic Flow Using Deep Spatiotemporal Neural Networks

Article *in* IEEE Transactions on Intelligent Transportation Systems · March 2020

DOI: 10.1109/TITS.2020.2979634

CITATIONS
33

READS
398

2 authors:

Tao Jia
Wuhan University
56 PUBLICATIONS **1,099** CITATIONS

SEE PROFILE

Penggao Yan
The Hong Kong Polytechnic University
5 PUBLICATIONS **87** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project shareability network View project

Project The restructure of retailing and the environmental impact View project

All content following this page was uploaded by Penggao Yan on 16 November 2020.

The user has requested enhancement of the downloaded file.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Predicting Citywide Road Traffic Flow Using Deep Spatiotemporal Neural Networks

Tao Jia and Penggao Yan

***Abstract*— Traffic flow forecasting has been a long-standing topic in intelligent transportation systems, and a renewed interest has been seen in recent years due to the development of artificial intelligence techniques. New deep neural networks have been developed to model traffic flow, but it is very challenging to predict citywide traffic flow at the road level in fine temporal scale owing to the influence of spatiotemporal dependencies and spatial sparsity. In this study, based on an in-depth analysis of traffic flow patterns, we propose a deep learning based spatiotemporal neural network model to predict citywide traffic flow for each road segment with high accuracy. Firstly, we examine the transformation of road network into its compact 2D image, where road segments correspond to pixels and their topological relationships are maintained in a large extent. Then, an end-to-end deep learning structure is designed to model traffic flow patterns. Specifically, recurrent convolutional network is employed to learn temporal dependencies and densely connected convolutional network is adopted to learn spatial dependencies and handle spatial sparsity. Our model attempts to aggregate the outputs of those hybrid networks using different weights, which is further enhanced by external information such as day of week. Experiments were conducted in Wuhan, China, where taxicab trajectory data were used to train and validate our model. When compared to current state of the art models, our model achieves higher accuracy in both single and multi-step traffic flow prediction tasks.**

***Index Terms*— Deep learning, spatiotemporal neural network, traffic flow prediction, trajectory data.**

I. Introduction

TRAFFIC flow forecasting is a long-standing topic in transportation research, and it is an indispensable component of intelligent transportation systems, helping to improve the efficiency of human travel, alleviate traffic congestion and pollutant emissions [1], and give government the ability to manage urban traffic smartly [2]. The main task of traffic flow forecasting is to predict the number of vehicles in a certain area or along a specified road for a future time period given historical traffic flow data [3], [4]. Advances in information and communication technologies have allowed the availability of an unprecedented amount of traffic data from road network, which are accumulated and constitute a massive historical dataset for developing and validating new methods or models [5]. Traditionally, real-time traffic data are collected by in-ground loop detectors or traffic surveillance camera systems, which are fixed measure stations that can obtain information in terms of flows, occupancy, speed, and videos [6]. However, it is relatively expensive to deploy and maintain the fixed monitoring system in a large area. Recently, the availability of massive location data, such as the trajectory data collected by GPS receivers installed on vehicles, have opened a new perspective to put forward novel predicting models using machine learning methods or deep learning models [3], [7].

In the literature, various machine learning methods have been proposed for traffic flow forecasting, which can be basically categorized into two classes: parametric approaches and nonparametric models. The parametric approaches make assumptions on the variables and establish a structural equation with parameters [8]. The most commonly used parametric models in traffic flow forecasting include time series based methods [9] and kalman filtering model [10]. For instance, two typical time series methods are autoregressive moving average method and autoregressive integrated moving average (ARIMA) method [11]. The parametric approaches are incapable of modeling the nonlinear property of traffic flow, and hence, much attention have been paid to the nonparametric models, which differ from parametric models in that the model structure is not specified as a priori but is instead determined by data. The most commonly used nonparametric models include k-nearest neighbor method [12], Bayesian network approach [13], support vector regression [14], and artificial neural networks [15]. For instance, reference [6] explored the short-term traffic prediction using a two network-based machine learning models considering spatiotemporal correlation among traffic variables. However, deficiencies of these methods might be their shallow structures, which cannot reveal hidden patterns of big traffic data.

To predict traffic flow effectively, many types of deep learning based methods have been proposed and explored in recent years [16]. Generally, using a multilayer nonlinear structure, deep learning model allows for an effective exploration of complex and nonlinear features hidden in big traffic data [2], [6], [9], [17]–[21]. Specifically, two main types of deep neural networks, convolutional neural networks (CNNs) and recurrent neural networks (RNNs), can capture very well the spatiotemporal structure of traffic flow [22]. CNNs are mainly used to learn the spatial dependencies, for instance, road traffic flow might be affected by its nearby roads and

Manuscript received April 20, 2019; revised August 24, 2019 and November 28, 2019; accepted February 24, 2020. This work was supported by the National Natural Science Foundation of China under Award 41971332. The Associate Editor for this article was L. Li. *(Corresponding author: Tao Jia.)*

The authors are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430072, China (e-mail: tao.jia@whu.edu.cn).

Digital Object Identifier 10.1109/TITS.2020.2979634

1524-9050 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Wuhan University. Downloaded on September 11,2020 at 08:10:19 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

distant roads, while RNNs are good at revealing the temporal dependencies, for instance, road traffic flow might be affected by the recent and previous time. As a specific type of RNNs, the long short-term memory network (LSTM) has been introduced to overcome the long-term dependencies problem [23] and used to predict traffic flow [4], [18], [19], [21], but spatial dependencies are not considered. Thereafter, a combination of CNN and LSTM are adopted to predict traffic flow by considering the spatiotemporal dependencies, but the experiment was conducted along only one single road [20]. Most recently, reference [22] proposed a deep spatiotemporal residual network (ST-Res) to predict citywide crowd flows, but prediction is at the region level without dealing with the challenges of spatial sparsity at the road level.

Therefore, this study aims to contribute a deep learning based spatiotemporal neural network to predict citywide traffic flow at the road level in fine temporal scale of 10 minutes with high accuracy. The major differences from the previous studies are focused on the following aspects. (1) We propose a method to transform the vector based road network into its compact 2D image, where each road segment corresponds to only one pixel and the topological relationships among road segments are maintained in a large extent. Specifically, it ensures the 2D image with the minimum size and the citywide prediction at the road level. (2) Based on an in-depth analysis of road traffic flow patterns, we develop a novel deep learning based spatiotemporal neural network, which is composed of four different parts accounting for modeling the spatiotemporal dependencies, spatial sparsity, and external factors. (3) Traffic flow is predicted at the road level for the entire city in very fine temporal scale of 10 minutes, which is a challenging issue and rarely investigated in the literature. The results can have practical implications for intelligent transportation systems. Experiments were conducted in Wuhan, China, where taxicab trajectory data were employed to train and validate our model. When compared to current state of the art techniques, our model achieves higher accuracy in both single and multi-step traffic flow prediction tasks.

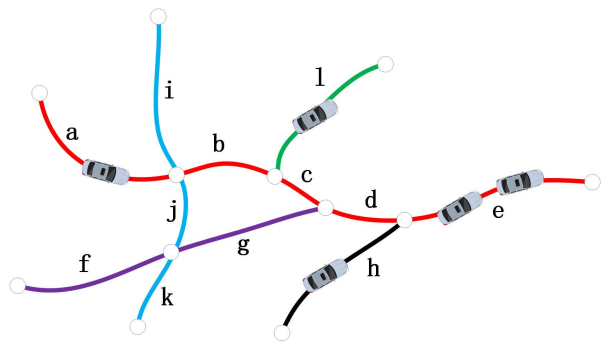
The remainder of this paper is organized as follows. In section 2, we describe the preliminaries of conceptual definition and the procedure to transform road network into its compact 2D image. In section 3, we propose detailed technique of the deep spatiotemporal neural network to predict traffic flow at the road level in fine temporal scale. In section 4, we present the experimental results and evaluate the accuracy compared with other methods. Model configuration and parameters setting are discussed in section 5. Conclusions are drawn in section 6.

II. Preliminary

A. Problem Statement

The main research problem is to predict road traffic flow RX_t during time period t given historical observations $\{RX_1, RX_2, \ldots, RX_{t-1}\}$. To tackle this problem, a few definitions are briefly introduced.

Traffic Flow: it is dynamic, complex, and constrained along a road segment or in a region [22]. A simple way to measure traffic flow is to count the number of vehicles on the road segment at certain time period. For instance, traffic flow on road segment i at time period t can be represented as rx_t^i.



Road ID	Flow	Road ID	Flow	Road ID	Flow
a	1	e	2	i	0
b	0	f	0	j	0
c	0	g	0	k	0
d	0	h	1	l	1

Fig. 1. Demonstration of road traffic flow at certain time period.

Road Traffic Flow: it can be defined as the traffic flow on all the roads at time period t, and it can be denoted as $RX_t = \{rx_t^i | i = 1, 2, \ldots, rn\}$, where rn is the number of road segments.

As shown in Fig.1, road traffic flow at certain time period is displayed, and it is zero if a road segment has no vehicles.

B. Transformation of Road Network Into Its Compact 2D Image

Different from the previous study of citywide traffic flow forecasting at the region level [22], we predict traffic flow at the road level. In this respect, we need to transform the road network into its compact 2D image, where each road segment corresponds to only one pixel. The transformation procedure attempts to maintain the topological relationships among the road segments in a large extent, which helps to avoid the loss of spatial dependencies and improve the accuracy of prediction; and it also tries to minimize the image size, which helps to alleviate the computational burden of model training. Specifically, the transformation procedure is composed of two steps.

As shown in Fig.2a, the first step is to represent road segments with their center points and divide their minimum boundary rectangle into spatial grids with optimal grid size iteratively using the Quadtree technique (please refer to **Algorithm 1** for the pseudocode). Here, the iteration times (k) is determined by road segments number (rn) and a given set of grid sizes ($grid_size$), and the optimal grid size is determined by maximizing the filling rate $rn/(grid_size^2 \times 4^k)$ such that the image size reaches the local minimum. The second step is to further divide each grid into individual cells according to the optimal grid size and map the road segments in each grid into the corresponding cells (please refer to **Algorithm 2** for the pseudocode). Here, the mapping process finds cell location for each road segment according to its relative position in the grid by ranking the center points with respect to x-coordinate

Authorized licensed use limited to: Wuhan University. Downloaded on September 11,2020 at 08:10:19 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

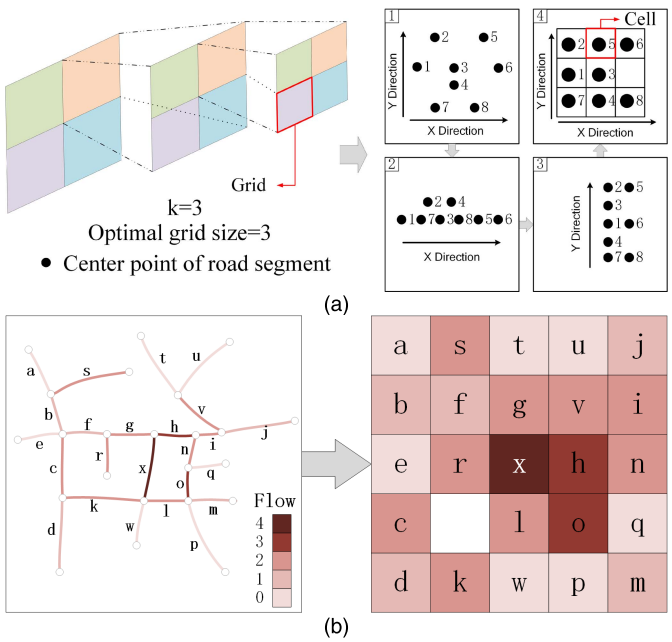


Fig. 2. An illustration of the procedure to transform the road network into its compact 2D image: (a) the procedure and (b) an example.

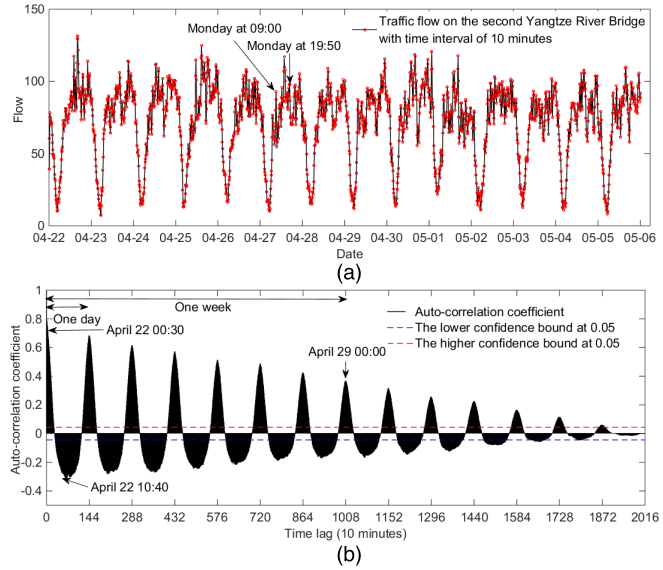


Fig. 3. Temporal dependencies of traffic flow at the road level: (a) the periodic pattern of traffic flow on the second Yangtze River Bridge and (b) the auto-correlation pattern of traffic flow with respect to different time lags.

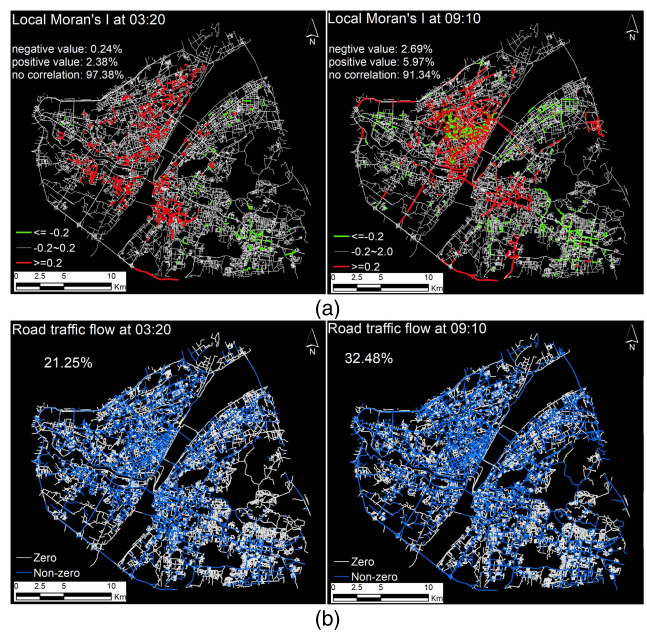


Fig. 4. Spatial dependencies of traffic flow at the road level: (a) local Moran's I at 03:20 and 09:10; (b) spatial sparsity of road traffic flow at 03:20 and 09:10, 27 April.

and y-coordinate. Specifically, an example is shown in Fig.2b, where a road network with 24 road segments is transformed into its compact 2D image with the size of 5×5 and the filling rate comes to 96%. In this respect, road traffic flow RX_t at time period t can be represented as $X_t = \{x_t^{p,q} | p = 1, 2, \ldots, m; q = 1, 2, \ldots, n\}$, where $x_t^{p,q} = rx_t^j$ if a road segment i is mapped into location *(p, q)* and $x_t^{p,q} = 0$ if no road segment is matched.

III. Deep Spatiotemporal Neural Networks

A. Spatiotemporal Patterns of Road Traffic Flow

It is well known that road traffic flow displays obvious spatiotemporal patterns. The temporal pattern can be examined through the temporal dependencies between the current traffic flow and the previous one. On one hand, traffic flow can display a periodic pattern within a region owing to the rhythm of human activities [22], [24]–[26], and a similar temporal pattern is reported at the road level in our study. As shown in Fig.3a, we can see clearly that traffic flow on the second Yangtze River Bridge shows a periodic temporal pattern for two weeks with peak times around 09:00 in the morning and 19:50 in the evening. On the other hand, temporal correlation of traffic flow can also display a periodic pattern, but it is gradually attenuating with time. Specifically, stronger correlation can be observed in a short time lag owing to the temporal dependencies, and weaker correlation can be expected in a long time lag owing to the temporal decay effect. As shown in Fig.3b, we can observe that traffic flow on the second Yangtze River Bridge on April 22 at 00:00 is better correlated with the one on April 22 at 00:30 (a time lag of 3) than the one on April 29 at 00:00 (a time lag of 1008). Interestingly, negative correlations can be also observed, for instance, traffic flow between 00:00 and 10:40. These temporal dependencies have been identified as major influencing factors for traffic flow forecasting, and hence, they should be integrated into our model.

The spatial pattern can be investigated from the spatial dependencies between traffic flow on current road segment and nearby road segments. Tobler's first law of geography states that near things are more related than distant things [27], and thus the spatial dependencies can be revealed using the technique of spatial auto-correlation. As shown in Fig.4a, spatial distributions of local Moran's I value are displayed for road traffic flow on April 22 at 03:20 and 09:10 respectively,

Authorized licensed use limited to: Wuhan University. Downloaded on September 11,2020 at 08:10:19 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

which suggest road traffic flow is much more correlated with each other in the peak time period (09:10) than in the nadir time period (03:20). Specifically, very few road segments (no more than 10%) show positive/negative correlations with their neighbors, whereas most of them display no correlation and are scattered around the city. As shown in Fig.4b, weak spatial correlation can be mainly explained by the sparse distribution of road traffic flow. For instance, 21% of road segments are occupied with vehicles at 03:20, while only 32% of road segments are occupied at 09:10. The percentages can still be decreased considering the filling rate in the image transformation procedure. These results hint that weak spatial correlation exists in road traffic flow and our model should be able to handle spatial sparsity of road traffic flow image and learn the spatial patterns.

B. Structure of the Network

Recurrent neural networks, such as LSTM or Gated-recurrent-unit network (GRU) [28], have been applied successfully to time series based prediction problem. However, they require a very long input sequence to train the model [16], [22], which is computationally expensive in our study. On the other hand, traditional CNNs have been reported to perform well on capturing image features, but they cannot learn the temporal patterns and spatial sparsity of images. Thus, based on an in-depth analysis of road traffic flow, we propose a deep learning based spatiotemporal neural network with the ability to learn the spatiotemporal dependencies and spatial sparsity of road traffic flow.

As shown in Fig.5a, the structure of our model is composed of four parts, namely recent part, daily part, weekly part, and external part. Recent part receives the input images of road traffic flow in a few previous steps (see Fig.5b, $\{X_{t-kr}, \ldots, X_{t-2}, X_{t-1}\}$, where kr is the length of recent input) and extracts the spatiotemporal features using a hybrid network of ConvLSTM [29] and DenseNet [30]. ConvLSTM network can extract the temporal dependencies effectively in recent time period, while DenseNet can extract the spatial dependencies effectively from the sparse image of road traffic flow. Daily part receives the input images of road traffic flow in one day before (see Fig.5b, $\{X_{(t-ld)\pm kd}, \ldots, X_{(t-ld)\pm 1}, X_{(t-ld)}\}$, where $2kd+1$ is the length of daily input, ld is the time period of one day and means 144 steps in terms of 10 minutes), and it adopts DenseNet to extract the spatial features of road traffic flow. Weekly part receives the input images of road traffic flow in one week before (see Fig.5b, $\{X_{(t-lw)\pm kw}, \ldots, X_{(t-lw)\pm 1}, X_{(t-lw)}\}$, where $2kw+1$ is the length of weekly input, lw is the time period of one week and means 1008 steps in terms of 10 minutes), and it uses a similar structure as the daily part. External part receives the date information (DayofWeek and Weekday/Weekend) as input and extracts the features using a two-layered fully connected network. The outputs from the first three parts are merged together by giving each output a different weight, and then the results are further combined with the output of the fourth part via an activation function to obtain predicted traffic flow X_t^{pre}.

1) The Structure of Recent Part: Recent part is mainly composed of two components, namely ConvLSTM and DenseNet.

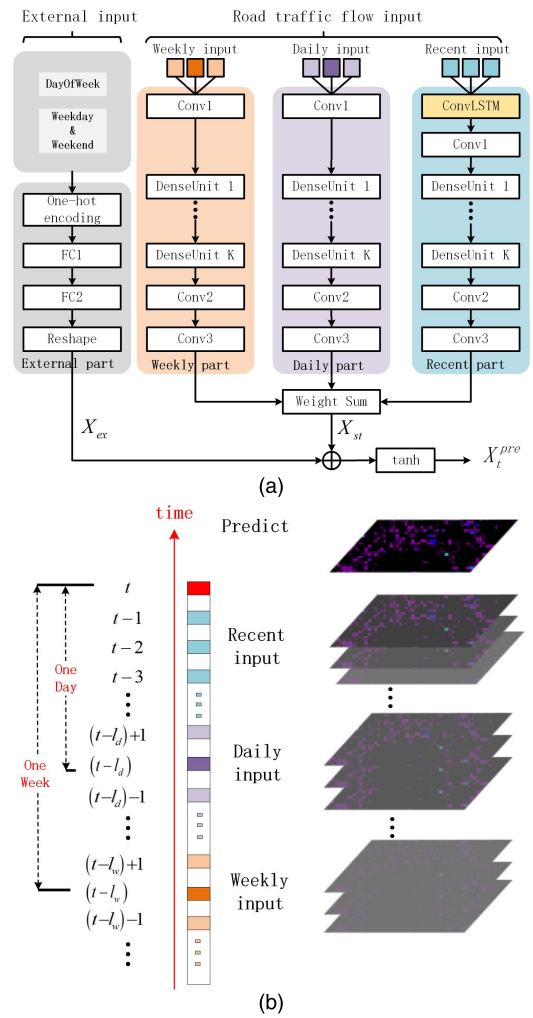


Fig. 5. An illustration of (a) the structure and (b) inputs of our model.

ConvLSTM [29] is different from traditional LSTM, where the input and recurrent transformations are convolutional. It is a specific type of RNNs capable of learning long short-term dependencies, and it has a cell-based chain-like structure, and each cell consists of four neural network layers interacted in a special way. It uses forget gate, input gate, and output gate to protect and control the state of cell. Forget gate can decide what information to be thrown away from the cell state, input gate can decide what new information to be stored in the cell state, and output gate can decide what information to be output.

Specifically, given an input sequence $\{X_{t-kr}, \ldots, X_{t-2}, X_{t-1}\}$, we can derive the output $\{h_{t-kr}, h_{t-kr+1}, \ldots, h_{t-1}\}$ from the ConvLSTM cell using (1), where $\Psi(\bullet)$ denotes inner operation in the cell, C_{t-kr-1} denotes initial state of the cell and is randomly initialized.

$$\begin{cases} h_{t-kr}, C_{t-kr} = \Psi\left(X_{t-kr}, C_{t-kr-1}, h_{t-kr-1}\right) \\ h_{t-kr+1}, C_{t-kr+1} = \Psi\left(X_{t-kr+1}, C_{t-kr}, h_{t-kr}\right) \\ \ldots\ldots \\ h_{t-1}, C_{t-1} = \Psi\left(X_{t-1}, C_{t-2}, h_{t-2}\right) \end{cases} \tag{1}$$

It should be noted that the last item h_{t-1} is chosen as the final output of ConvLSTM, which is then fed into Batch

Authorized licensed use limited to: Wuhan University. Downloaded on September 11,2020 at 08:10:19 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Normalization (BN) [31] and Conv1 layer. BN can accelerate deep network training by reducing internal covariance shift, and Conv1 is a convolutional layer as shown in (2), where W_{conv1} and b_{conv1} are learnable parameters, and M_{r1} is the output that will be fed into DenseNet elaborated below.

$$M_{r1} = W_{conv1}{}^{*}h_{t-1} + b_{conv1} \tag{2}$$

DenseNet [30] is a specific type of CNNs that creates short paths from early layers to later layers to alleviate the vanishing-gradient problem, and in this respect, it can substantially reduce the number of parameters. Compared with the well-known ResNets [22], it does not combine features using summation but concatenating them. Most importantly, DenseNet is more capable of extracting spatial features from sparse road traffic flow image than ResNets. Dense unit is the basic structure of DenseNet, and each unit consists of several convolutional layers interacting in a special way. Each convolutional layer is a BRConv, which consists of a BN layer, a rectified linear unit layer (where an activation function ReLU is used), and a convolutional layer (Conv) in sequence. For the sake of convenience, we use X_0 instead of M_{r1} as the input, and the structure of a dense unit can be represented in (3), where l is BRConv number in the dense unit, $Con[\bullet]$ denotes concatenation of elements, and $H_l(\bullet)$ is BRConv operation.

$$\begin{cases} X_1 = H_1(X_0) \\ X_2 = H_2(Con[X_1, X_0]) \\ \cdots\cdots \\ X_l = H_l\left(Con\left[X_{l-1}, \ldots, X_1, X_0\right]\right) \\ Y_l = Con\left[X_l, X_{l-1}, \ldots, X_1, X_0\right] \end{cases} \tag{3}$$

Particularly, DenseNet is composed of a sequence of connected dense units, and we can derive the output Y_l^k from DenseNet with k dense units using (4), where g is the growth rate of the network [30].

$$\begin{cases} Y_l^1 = \Phi(X_0, g, l) \\ Y_l^2 = \Phi\left(Y_l^1, g, l\right) \\ \cdots\cdots \\ Y_l^k = \Phi\left(Y_l^{k-1}, g, l\right) \end{cases} \tag{4}$$

Finally, two convolutional layers, Conv2 and Conv3 as shown in Fig.5a, are applied to the output of DenseNet, and they can be represented in (5), where W_{conv2}, b_{conv2}, W_{conv3}, and b_{conv3} are all learnable parameters, *ReLu(•)* is a rectified linear transform function and M_{r3} is the final output of recent part.

$$\begin{cases} M_{r2} = W_{conv2}{}^{*}Y_l^k + b_{conv2} \\ M_{r3} = W_{conv3}{}^{*}ReLu(M_{r2}) + b_{conv3} \in R^{1\times m\times n} \end{cases} \tag{5}$$

2) The Structure of Daily and Weekly Parts: The structure of daily and weekly parts are similar to recent part except that there is no ConvLSTM. Thus, we directly use convolutional operation in (2) to obtain the output M_{d1}, which is then fed into DenseNet with k dense units in (4) to derive output Y_l^k. Finally, Equation (5) is used to calculate outputs of M_{d2} and M_{d3}, and M_{d3} is the final output of daily part. Similarly, we can derive the final output of weekly part as M_{w3}.

3) The Structure of External Part: It is well known that traffic flow can be slightly different from one day to another or particularly from weekday to weekend. To achieve a high accurate traffic flow forecasting, we integrate the date information, such as DayofWeek and Weekday/Weekend, into our model as external part. Although other external information such as weather and accident might affect traffic flow, they are not adopted because weather conditions remain unchanged during the time period and accident data are unavailable. Thus, a one-hot eight-bit representation is used to encode date information, where the first seven codes denote the day of week and the last code labels information of Weekday or Weekend, for instance, {1,0,0,0,0,0,0,1} indicates Monday in Weekday and {0,0,0,0,0,1,0,0} indicates Saturday in Weekend. Secondly, we adopt two stacked fully-connected layers (FC1 and FC2). FC1 is used to extract hidden features of date information, while FC2 is only used to expand the size of output vector from FC1 such that it has the same size as the flattened M_{r3}. Finally, the output vector is further reshaped and denoted as $X_{ex} \in R^{1\times m\times n}$.

4) Fusion: Two fusions are conducted in our model. The first fusion is to adopt a parametric-matrix-based fusion method [22] to combine the outputs from recent part, daily part, and weekly part, which is named as the Weight Sum layer in Fig.5a and is represented in (6), where W_r, W_d, and W_w are learnable parameters that can give different weights to the three parts.

$$X_{st} = W_r \circ M_{r3} + W_d \circ M_{d3} + W_w \circ M_{w3} \in R^{1\times m\times n} \tag{6}$$

The second fusion is to combine the outputs from external part (X_{ex}) and the first fusion output (X_{st}), and they are directly added together as shown in Fig.5a. To ensure the output values between -1 and 1, we apply a tanh function to the output, and it is represented in (7), where X_t^{pre} is the predicted road traffic flow at time period t.

$$X_t^{pre} = tanh(X_{st} + X_{ex}) \in R^{1\times m\times n} \tag{7}$$

C. Training and Evaluation of the Network

1) Network Training: To train our model, we firstly derive road traffic flow images for the three input sequences from massive historical traffic flow data. Secondly, using forward and backward propagation and Adam method [32], the input images are gradually fed into our model with the aim to minimize the mean squared error $L(\Theta)$ between X_t^{pre} and X_t (as shown in (8), Θ are all learnable parameters). Thirdly, we obtain the well-trained model Λ when the total mean square error is significantly small.

$$L(\Theta) = \|Xt - X_t^{pre}\|^2 \tag{8}$$

It should be noted that model Λ is designed for single step road traffic flow prediction task. However, for multi-step prediction task, we reconstruct the three sequences of input images using result of single step ahead prediction. Detailed algorithms for model training and prediction is given in **Algorithm 3** and **Algorithm 4** for the pseudocode.

Authorized licensed use limited to: Wuhan University. Downloaded on September 11,2020 at 08:10:19 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

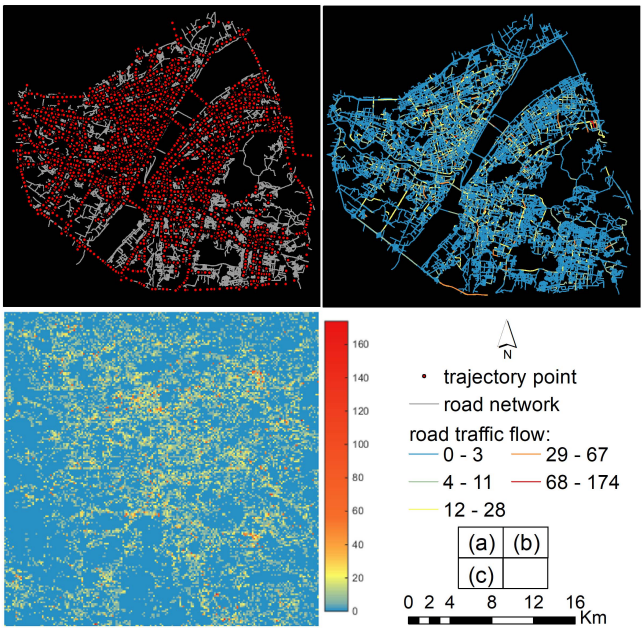


Fig. 6. Data used in our study: (a) taxicab trajectories overlaid on road network; (b) observed traffic flow; and (c) derived traffic flow image.

2) Network Evaluation: To evaluate the trained model, we adopt two metrics to evaluate its accuracy, namely the mean absolute error *MAE* in (9) and the root mean square error *RMSE* in (10).

$$MAE = 1/\mathrm{N} \sum_i \left| X_i - X_i^{pre} \right| \tag{9}$$

$$RMSE = \sqrt{1/\mathrm{N} \sum_i \left(X_i - X_i^{pre} \right)^2} \tag{10}$$

where X_i is the ith observed value, X_i^{pre} is the ith predicted value, and N is the number of observations.

IV. Experimental Results and Analysis

A. Data and Preprocessing

In this study, taxicab GPS trajectory data within the third-ring-road in Wuhan, China (see Fig.6a) were employed to calculate observed road traffic flow. The data can provide partial information, but they are ubiquitous and have been used intensively in traffic flow studies [6], [22], [25]. A total number of 195,434,506 trajectory points were contributed by about 18,600 taxicabs from April 8 to May 5 in 2015. Each trajectory point is allocated to the nearest road segment, and this procedure is conducted for all trajectory points to derive road traffic flow at time interval of 10 minutes (see Fig.6b).

To obtain network input image, road traffic flow is transformed into its compact 2D image using our method as introduced in section 2.2. As shown in Fig.6c, we can clearly see that a compact 2D image is derived from road traffic flow data, where traffic flow of each road segment is assigned to the value of the corresponding pixel. Statistically, our road network has a total number of 35,575 road segments, the derived traffic flow image is composed of 36,864 (192 × 192) pixels, and thus the filling rate is as high as 96.5%. Besides, we use the Min-Max normalization method to scale pixel values of input images into the range between −1 and 1 to satisfy the requirement of our model. As a reverse calculation, we can re-scale the predicted traffic flow value back to the normal value. In addition, we use the one-hot eight-bit representation technique to encode date information to provide external inputs as introduced in section 3.2.

TABLE I
Parameter Settings of Our Model

Layers	Output	Recent part	Daily part	Weekly part
ConvLSTM	192 x 192	3 x 3, 8	\	\
BN				
Conv1	192 x 192	3 x 3, 16	3 x 3, 16	3 x 3, 16
Dense Unit	192 x 192	$\begin{bmatrix} 1 \times 1 \text{ BRConv} \\ 3 \times 3 \text{ BRConv} \end{bmatrix} \times 8$	$\begin{bmatrix} 1 \times 1 \text{ BRConv} \\ 3 \times 3 \text{ BRConv} \end{bmatrix} \times 8$	$\begin{bmatrix} 1 \times 1 \text{ BRConv} \\ 3 \times 3 \text{ BRConv} \end{bmatrix} \times 8$
Conv2	192 x 192	3 x 3, 8	3 x 3, 8	3 x 3, 8
ReLU				
Conv3	192 x 192	3 x 3, 1	3 x 3, 1	3 x 3, 1
Weighted Sum Layer, Tanh				

B. Model Settings

Firstly, as shown in TABLE I, we elaborate parameter settings of our model, which are related to structures of ConvLSTM, Conv1, DenseNet with one dense unit, Conv2, and Conv3. ConvLSTM uses filters of 3 × 3 to conduct input and recurrent transformations and output 8 feature-maps of 192 × 192, and it only works for recent part; Conv1 uses 16 filters of size 3 × 3, and it works for recent part, daily part, and weekly part; Each dense unit has 8 BRConv layers, whose convolution part uses filters of 3 × 3 and growth rate (g) of 4, and it works for the three parts; Conv2 uses 8 filters of size 3 × 3, while Conv3 uses only 1 filter of size 3 × 3. Both of the two convolutional layers work for the three parts. Besides, Adam method [32] is used for optimization, and batch size is set as 8.

Secondly, we introduce parameter settings on network input, and there are three parameters that need to be specified including *kr*, *kd*, and *kw*. Specifically, *kr* is the parameter of input sequence for recent part and is set as 3; *kd* is for daily part and is set as 1; *kw* is for weekly part and is set as 1. It should be noted that all parameter settings will be discussed in section 5. In addition, the observed traffic flow images are demarcated into two sets: the training set and the testing set. About 60% of data is used for training, while the remaining 40% is used for testing. Among the training set, 90% of data is used for model training and 10% is used for cross validation.

C. Results of the Predicted Road Traffic Flow

As shown in Fig.7a, we present spatial distribution of the predicted road traffic flow at 09:10 from 09:00 on April 27 2015. Results suggest that some major urban roads are occupied with heavy traffic flow owing to the commuting behavior in early morning peak time, such as the Luoyu road

Authorized licensed use limited to: Wuhan University. Downloaded on September 11,2020 at 08:10:19 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

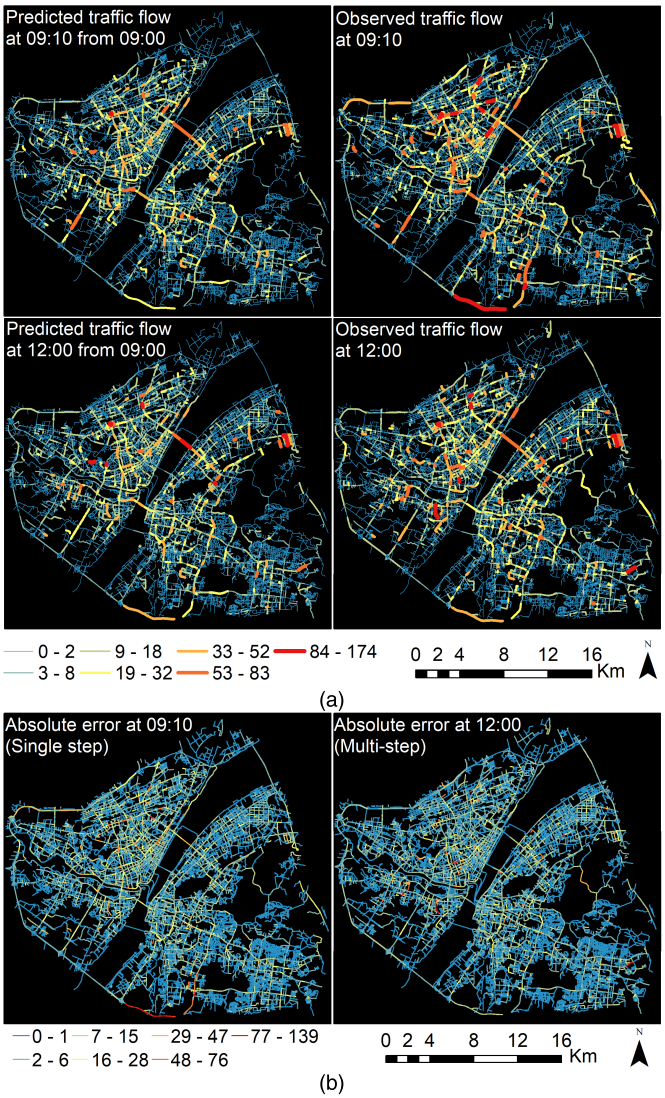


Fig. 7. Comparison of predicted road traffic flow with the observed one: on April 27, 2015: (a) at 09:10 and 12:00; (b) absolute error at 09:10 and 12:00.

in Wuchang and the second Yangtze River Bridge connecting Hankou and Wuchang. Importantly, traffic flow in most of the urban roads has been predicted very well at 09:10 and is in high consistence with that observed. However, traffic flow in road segments along the third-ring-road deviates a little bit from that observed. This is mainly due to the edge or boundary effects in ecology, which states that traffic flow outside the boundary might have some influences on these along the boundary. Statistically, as shown in Fig.7b, only 0.06% of road segments display very large values of absolute error (>50).

In addition, as shown in Fig.7a, we display spatial distribution of the predicted road traffic flow at 12:00 from 09:00 on April 2015. Specifically, traffic flow is predicted three hours later, which means single step ahead prediction is repeatedly conducted 18 times. Overall, compared with the observed value, traffic flow predicted at 12:00 is worth than that predicted at 09:10, although most of urban major roads can still be predicted very well. As shown in Fig.7b,

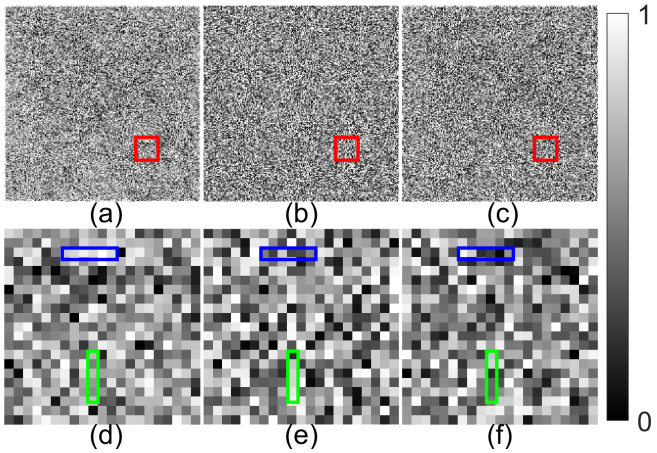


Fig. 8. Visualization of influences on the prediction for (a) recent part, (b) daily part, (c) weekly part, and the enlarged area (red box) in (d) recent part, (e) daily part, (f) weekly part.

about 0.08% of road segments display very large values of absolute error (>50) owing to error propagation. These road segments with large errors are scattered in urban area, and their traffic flows are more likely to be underestimated. This result indicates that our model can be robust for road traffic flow prediction in multi-step task.

The above results show that our model could predict citywide road traffic flow in fine temporal scale of 10 minutes with high accuracy. However, it is unknown how recent part, daily part, and weekly part of our model affect the predicted traffic flow. As shown in Fig.8a, 8b, and 8c, we visualize the parametric matrix of Weight Sum Layer and display the influence of recent part, daily part, and weekly part on the predicted traffic flow, respectively. A general statistical analysis suggests that influence of recent part is much stronger than the other two parts. For instance, if weight value is set as 0.6, there is around 48% of road segments affected by recent part, which is relatively greater than 45% and 41% affected by daily part and weekly part, respectively. Specifically, some road segments might be more affected by recent part, while others might be more affected by the other two parts. For instance, as shown in Fig.8d, 8e, and 8f, road segments enclosed by blue box are more affected by recent part, while those enclosed by green box are more affected by daily part.

D. Evaluation of the Results

To evaluate our model, results were compared with the state of the art forecasting methods, which includes Vector Auto-Regressive (VAR), ARIMA, LSTM, GRU, and ST-Res.

VAR is a stochastic process model used to capture linear interdependencies among multiple time series. In this study, we flatten our 2D flow image to 1D sequence as model input.

ARIMA can be applied to time series data either for better understanding the data or for predicting future points. Again, the 2D flow image is flattened into 1D sequence as input.

LSTM [23] can learn the temporal dependencies of long-term time series data and thus can be used to make predictions. In this study, the length of input (flow image)

Authorized licensed use limited to: Wuhan University. Downloaded on September 11,2020 at 08:10:19 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II

COMPARISON OF OUR MODEL WITH THE STATE OF THE ART MODELS (NOTE: ○ INDICATES NO AND ● INDICATES YES)

Model	Spatial dependencies	Temporal dependencies	External factors	Spatial sparsity
VAR	○	●	○	○
ARIMA	○	●	○	○
LSTM-9/12/16	●	●	○	○
GRU-9/12/16	●	●	○	○
ST-Res	●	●	●	○
Our model	●	●	●	●

sequence is set as 9, 12, and 16, which corresponds to LSTM-9, LSTM-12, and LSTM-16, respectively.

GRU [28] is a new kind of RNN and is particularly suitable for modeling long-term temporal dependencies. Being the same as the setting of LSTM, we use GRU-9, GRU-12, and GRU-16 in the experiment.

ST-Res [22] is a newly proposed model for learning the spatiotemporal dependencies of time series data, but it is designed to predict traffic flow at the region level. In this study, our road traffic flow images are used as network input, and other settings are the same as the previous study [22].

As shown in TABLE II, we list characteristics of these models. Specifically, ARIMA has the same capacity as VAR, which can only capture temporal dependencies. LSTM and GRU can learn spatial and temporal dependencies, but they do not consider external factors. ST-Res and our model have the same ability to capture both spatiotemporal dependencies and external factors, but our model can further deal with spatial sparsity of traffic flow image. These models were implemented under the same computing environment and have different time costs (please refer to Table III for the detailed descriptions).

Comparative results are shown in Fig.9, where we display values of *MAE* and *RSME* with respect to different models for evaluating the results of single step ahead prediction and multi-step ahead prediction. As shown in Fig.9a and 9b, accuracies of different models in terms of *MAE* are consistent with the values of *RSME*. Particularly, ARIMA and VAR perform the worst owing to their incapacity of capturing spatial dependencies; while our model performs the best in terms of *MAE*. As for *RSME*, it does not deviate too much from the best model GRU-9, which still suggests the effectiveness of our model. This result demonstrates that spatiotemporal dependencies can impose a heavy influence on single step ahead prediction, and thus the models capable of learning spatiotemporal dependencies do not deviate too much from each other in terms of the accuracy. However, as shown in Fig.9c and 9d for multi-step ahead prediction, our model seems to perform the best among all models. Specifically, as the number of look-ahead step increases, values of *MAE* and *RMSE* are gradually increased for LSTM or GRU variants, because the length of current input sequence is too short to support long-term forecasting. Meantime, accuracy of our model and the ST-Res are relatively stable with the increment

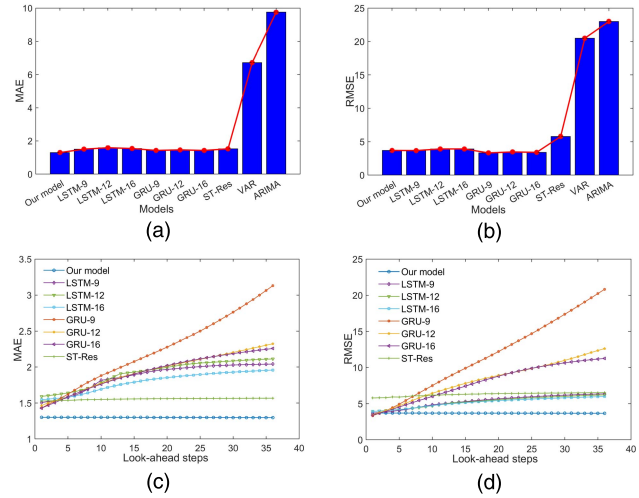


Fig. 9. Accuracy comparison of our model with other methods in road traffic flow forecasting: single step ahead prediction using (a) *MAE* and (b) *RMSE*; multi-step ahead prediction using (c) *MAE* and (d) *RMSE*.

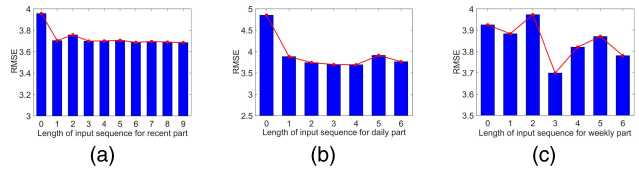


Fig. 10. Impact of length of network input sequence: (a) kr; (b) $2kd + 1$; and (c) $2kw + 1$.

of the number of look-ahead step, because both of them can learn the periodical pattern of traffic flow. However, values of *MAE* and *RMSE* of our model are on average 16.6% and 41.4% lower than those of ST-Res, respectively. This is because our model is more adept at learning spatial patterns from the sparse images. Thus, comparative results suggest that our model is effective and reliable for both single and multi-step traffic flow prediction tasks.

V. DISCUSSIONS

A. Impact of Network Input

The length of network input sequence can affect the prediction. In this study, three parameters, namely kr, kd, and kw, are related to the input sequence of recent part, daily part, and weekly part, respectively. As for kr shown in Fig.10a, we can see that our model performs the best when it is set as 3 (the input length is 3), although very slight differences can be observed for different values. As for kd shown in Fig.10b, we can observe that our model performs the best when it is set as 1 (the input length is $2kd + 1 = 3$), which suggests that a long input sequence for daily part could not always improve the accuracy. Specifically, the above two parameters cannot be ignored, because prediction accuracy can be decreased when their values are set as 0. As for kw shown in Fig.10c, we can find that our model performs the best when it is set as 1 (the input length is $2kw + 1 = 3$). The impact of weekly part is weaker than the other two parts, but it cannot be neglected. These results imply that the length of network input should be carefully treated.

Authorized licensed use limited to: Wuhan University. Downloaded on September 11,2020 at 08:10:19 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

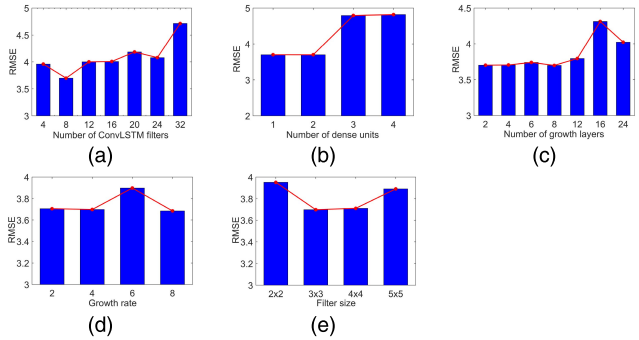


Fig. 11. Impact of network structure: (a) filters number; (b) dense unit number; (c) growth layers number; (d) growth rate; and (e) filter size.

B. Impact of Network Structure

1) Structure of ConvLSTM: ConvLSTM is only used in recent part. A very important parameter in ConvLSTM is filters number, which reflects the dimension of output features. As shown in Fig.11a, we display the change of filters number with the accuracy in terms of *RMSE*. At the beginning, *RMSE* decreases with the increment of filters number, which is reasonable because high dimensions of output features might contain rich spatiotemporal information. However, *RMSE* increases with more filters are included. This is probably due to redundant information, which might increase the burden of model training. Therefore, it can be set as 8, which results in a relatively small value of *RMSE*.

2) Structure of DenseNet: DenseNet is used in all the three parts. Three parameters, including dense unit number (k), BRConv layers number (l), and growth rate (g), are related to the structure of DenseNet. As shown in Fig.11b, *RMSE* does not decrease with the increment of dense unit number, which suggests that one dense unit is enough to capture the spatial patterns. As shown in Fig.11c, dense unit becomes much deeper with the increasing value of l, but *RMSE* does not decrease continuously as assumed. Instead, it fluctuates around 3.7 when l increases from 2 to 12, and finally a local minimum is achieved when l is set as 8. In addition, as shown in Fig.11d, we set the value of g as 4 because a local minimum of *RSME* is reached.

3) Filter Size: Filter size determines the receptive filed of a convolutional operation. As shown in Fig.11e, change in filter size has little effect on *RMSE*. This is because a filter with large size might be equal to a combination of many filters with small size, which could extract a similar spatial pattern. Specifically, *RMSE* reaches a local minimum when the filter size is 3×3, which might be related to the spatial auto-correlation patterns of road traffic flow. Indeed, as shown in Fig.4, weak local spatial auto-correlation can be captured by a filter with small size.

C. Impact of External Factors

External factors such as DayofWeek and Weekday/Weekend have been considered in our model, but there are many other factors, such as holidays, weather, and accident, which could affect our prediction. A simple comparison of prediction accuracy on holidays and workdays found that factor of holidays has a limited influence, e.g., prediction accuracy in terms of *RMSE* is 3.725 and 3.721 for workday (29 April) and holiday (1 May), respectively. However, factors of weather and accident are not considered, because weather conditions keep stable and accident data are unavailable. This is a limitation of our study and will be integrated into our model in future.

Algorithm 1 Division of Road Network Using Quadtree

Input:
(1) The center points of road segments P and the number N;
(2) Candidate set of grid size $sGrid = \{2, 3, 5\}$;
//Step 1: Find the best grid size grid_size
Initialize unfilled ratio set $uRatio = \{0, 0, 0\}$, iteration set $sIterat = \{0, 0, 0\}$;
For $i = 0$ **to** *length(sGrid)*-1
  Find maximum k s.t. $sGrid[i]^2 \times 4^{k-1} \leq N \leq sGrid[i]^2 \times 4^k$;
  $sIterat[i] = k$, $uRatio[i] = 1 - N/(sGrid[i]^2 \times 4^k)$;
End
Find the index of the minimum value in *uRatio* as *i_grid*;
$grid_size = sGrid[i_grid]$, $iterations = sIterat[i_grid]$;
//Step 2: Iteration process
Initialize segmentation set as $S = \{P\}$;
For $j = 1$ **to** *iterations*
  For each subset P **in** S
    Find the center point of P in x direction as p_x;
    $P_1 = \{p|\ p.lon < p_x.\ lon,\ p. \in P\}$, $P_2 = \{p|\ p.lon > p_x.\ lon,\ p. \in P\}$; $P_2 = \{p|p.lon > p_x.lon, p \in P\}$
    Find the center point of P_1 in y direction as p_{y1};
    $P_a = \{p|\ p.lat < p_{y1}.\ lat,\ p. \in P_1\}$, $P_b = \{p|\ p.lat > p_{y1}.\ lat,\ p. \in P_1\}$; $P_b = \{p|p.lat > p_{y1}.lat, p \in P_1\}$
    Find the center point of P_2 P_2 in y direction as p_{y2};
    $P_c = \{p|\ p.lat < p_{y2}.\ lat,\ p. \in P_2\}$, $P_d = \{p|\ p.lat > p_{y2}.\ lat,\ p. \in P_2\}$;
    Insert P_a, P_b, P_c, P_d into S_{tmp};
  End
  $S = S_{tmp}$, $S_{tmp} = \emptyset$;
End
Output: S

D. Impact of the Compact 2D Image

The Organization of compact 2D image can affect the prediction accuracy. This study proposed a novel method to organize the compact 2D image such that topological relationships can be maintained in a large extent. To verify the effectiveness of our conversion, experiments were conducted to compare our results with those obtained from a

Authorized licensed use limited to: Wuhan University. Downloaded on September 11,2020 at 08:10:19 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 2 Mapping Road Segment to the Corresponding Cell

Input: The segmentation set S, the grid size *grid_size*;
Initialize matrix set as γ *_set*;
For each subset Q in S
    Sort points of Q in x-direction;
    Initialize matrix γ of size *grid_size* x *grid_size*;
    For $i = 0$ to *grid_size* - 1
        $Q_{tmp} = Q[i\colon (i+1)$ x *grid_size*];
        Sort points of Q_{tmp} in y-direction;
        $\gamma[:][i] = Q_{tmp}$;
    End
    Insert γ into γ *_set* and set γ as $\emptyset$;
End
Reorganize γ *_set* into 2D image X;
Output: X

Algorithm 3 Training Our Model

Input:
(1) Historical traffic flow image $\{X_0, X_1, \ldots, X_{n-1}\}$;
(2) External features $\{E_{x,0}, E_{x,1}, \ldots, E_{x,n-1}\}$;
(3) Lengths of input sequences kr, $2kd + 1$, $2kw + 1$;
(4) Daily lag ld and weekly lag lw;
//Step 1: Feature organization
For $t = 1$ to n-1
    $X_r = [X_{t-kr}, \ldots, X_{t-2}, X_{t-1}]$, $X_d = [X_{(t-ld)\pm kd}, \ldots, X_{(t-ld)\pm 1}, X_{(t-ld)}]$;
    $X_w = [X_{(t-lw)\pm kw}, \ldots, X_{(t-lw)\pm 1}, X_{(t-lw)}]$, $X = [X_r, X_d, X_w, E_{x,t}]$;
    Insert (X, X_t) into $\mathcal{T}$;
End
//Step 2: Model training and optimization
Initialize model parameters randomly as Θ and total loss as $Loss = 0$;
Repeat
    For each(X, X_t)**in** $\mathcal{T}$
        Perform forward propagation to compute X_t^{pre};
        Compute mean square error $L(\Theta) = \|X_t - X_t^{pre}\|^2$;
        $Loss += L(\Theta)$;
        Perform backward propagation to compute $\Delta\Theta$ and $\Delta\Theta'$;
        $\Theta = \Theta + \Delta\Theta$, $\Theta' = \Theta' + \Delta\Theta'$;
    End
Until $Loss < \sigma$;
Output: Trained model Λ

random organization. The random matching conversion simply assigns a road segment to a randomly selected unoccupied image pixel, and it continues until all the road segments have been mapped to the corresponding image pixels. In this way, the topological relationships can be lost totally. Results suggest that, using our method, prediction accuracy can be improved by around 6% in terms of *RMSE* for single step ahead prediction.

Algorithm 4 Multi-Step Ahead Prediction Using Our Model

Input:
(1) Trained Model Λ, number of look-ahead steps s;
(2) Historical traffic flow image $\{X_0, X_1, \ldots, X_{n-1}\}$;
(3) External features $\{E_{x,n}, E_{x,n+1}, \ldots, E_{x,n+s-1}\}$;
$H \leftarrow \{X_0, X_1, \ldots, X_{n-1}\}$;
For $t = n$ to $n + s - 1$
    $X_r = [H_{t-kr}, \ldots, H_{t-2}, H_{t-1}]$, $X_d = [H_{(t-ld)\pm kd}, \ldots, H_{(t-ld)\pm 1}, H_{(t-ld)}]$;
    $X_w = [H_{(t-lw)\pm kw}, \ldots, H_{(t-lw)\pm 1}, H_{(t-lw)}]$;
    Predict $X_t^{pre} \leftarrow \Lambda\{X_r, X_d, X_w, E_{x,t}\}$;
    Append X_t^{pre} into H;
End
Output: $\{X_n^{\text{pre}}, X_{n+1}^{\text{pre}}, \ldots, X_{n+s-1}^{\text{pre}}\}$

TABLE III
TIME COSTS OF OUR EXPERIMENTS USING DIFFERENT MODELS

(Note: Experiments were conducted under the same computing environment, which refers to a high performance computer with the configuration of a RAM of 32GB, an Intel Core i7-7800X processor with 6 cores of 3.50 GHz, and a NVIDIA GeForce GTX 1080 Ti Graphic Card of 11 GB.)

Model	Compiling time (s)	Training Time (s)	Testing Time (s)
LSTM-9	17.62	10282.09	80.82
LSTM-12	17.75	13586.15	87.38
LSTM-16	19.73	16013.28	100.25
GRU-9	18.64	9981.32	73.51
GRU-12	20.65	12105.72	82.89
GRU-16	18.53	14102.47	95.66
ST-Res	20.48	48114.88	132.57
Our Model	**18.12**	**9641.71**	**115.84**

VI. CONCLUSION

This study proposed a deep learning based spatiotemporal neural network to predict citywide traffic flow at the road level in fine temporal scale with high accuracy. We firstly transform road network into its compact 2D image such that topological relationships can be maintained in a large extent. Then, according to spatiotemporal patterns of traffic flow, the structure of our network is designed with four parts including recent, daily, weekly, and external parts. The first three parts are responsible for capturing the spatiotemporal dependencies of sparse images, while the last part can learn external information such as day of week. Lastly, outputs from the four parts are fused together with weights learned during the training process. Specifically, parameters of our network model are carefully treated and visualized for making a high accurate prediction.

Experiments were conducted in Wuhan, China. Trajectory points were assigned to road network to obtain traffic flow, which were then transformed into its compact 2D image. Based on these images, our model was trained and the network parameters were empirically determined. The predicted city-wide road traffic flows in 10 minutes were visually analyzed in both single and multi-step prediction tasks, and they agree very well with those observed. Particularly, we found that they are much more affected by recent part than the other two parts. Importantly, compared with those predicted by the state of the

Authorized licensed use limited to: Wuhan University. Downloaded on September 11,2020 at 08:10:19 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

art methods, the results indicate that our model is effective and reliable for both single step and multi-step citywide traffic flow prediction with very high accuracy in 10 minutes. The Python code of our model and the traffic flow data are available on our GitHub repository (https://github.com/JasonYanxx/STNN).

Appendix

See Algorithms 1–4 and Table III.

Acknowledgment

The authors would like to thank the editor and anonymous referees for their valuable comments for improving the manuscript.

References

[1] T. Jia, Q. Li, and W. Shi, "Estimation and analysis of emissions from on-road vehicles in Mainland China for the period 2011–2015," *Atmos. Environ.*, vol. 191, pp. 500–512, Oct. 2018.

[2] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 1–17, Jun. 2017.

[3] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. C, Emerg. Technol.*, vol. 43, no. 1, pp. 3–19, Jun. 2014.

[4] J. Mackenzie, J. F. Roddick, and R. Zito, "An evaluation of HTM and LSTM for short-term arterial traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1847–1857, May 2019.

[5] Q. Shi and M. Abdel-Aty, "Big data applications in real-time traffic operation and safety monitoring and improvement on urban expressways," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 380–394, Sep. 2015.

[6] G. Fusco, C. Colombaroni, and N. Isaenko, "Short-term speed predictions exploiting big data on large urban road networks," *Transp. Res. C, Emerg. Technol.*, vol. 73, pp. 183–201, Dec. 2016.

[7] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transport. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[8] S. Mazahir, O. Hasan, R. Hafiz, and M. Shafique, "Probabilistic error analysis of approximate recursive multipliers," *IEEE Trans. Comput.*, vol. 66, no. 11, pp. 1982–1990, Nov. 2017.

[9] T. Thomas, W. Weijermars, and E. Van Berkum, "Predictions of Urban volumes in single time series," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 71–80, Mar. 2010.

[10] Z. Su, A. Jamshidi, A. Núñez, S. Baldi, and B. De Schutter, "Multi-level condition-based maintenance planning for railway infrastructures–A scenario-based chance-constrained approach," *Transp. Res. C, Emerg. Technol.*, vol. 84, no. 1, pp. 92–123, Nov. 2017.

[11] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *Eur. Transp. Res. Rev.*, vol. 7, no. 3, pp. 1–9, Sep. 2015.

[12] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An improved K-nearest neighbor model for short-term traffic flow prediction," *Procedia-Social Behav. Sci.*, vol. 96, pp. 653–662, Nov. 2013.

[13] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 124–132, Mar. 2006.

[14] J. Wang and Q. Shi, "Short-term traffic speed forecasting hybrid model based on Chaos–Wavelet analysis-support vector machine theory," *Transp. Res. C, Emerg. Technol.*, vol. 27, pp. 219–232, Feb. 2013.

[15] T. Ma, Z. Zhou, and B. Abdulhai, "Nonlinear multivariate time–space threshold vector error correction model for short term traffic state prediction," *Transp. Res. B, Methodol.*, vol. 76, pp. 27–47, Jun. 215.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[17] H. Tan, Y. Wu, B. Shen, P. J. Jin, and B. Ran, "Short-term traffic prediction based on dynamic tensor completion," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2123–2133, Aug. 2016.

[18] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, Nov. 2016, pp. 324–328.

[19] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "LSTM-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297–305, Nov. 2018.

[20] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transp. Res. C, Emerg. Technol.*, vol. 90, pp. 166–180, May 2018.

[21] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Proc. IEEE Int. Conf. Smart City/SocialCom/SustainCom (SmartCity)*, Chengdu, China, Dec. 2015, pp. 153–158.

[22] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artif. Intell.*, vol. 259, pp. 147–166, Jun. 2018.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] T. Jia and B. Jiang, "Exploring human activity patterns using taxicab static points," *ISPRS Int. J. Geo-Inf.*, vol. 1, no. 1, pp. 89–107.

[25] Z. Hu, T. Jia, G. Wang, J. Wang, and L. Meng, "Computing a hierarchy favored optimal route in a Voronoi-based road network with multiple levels of detail," *Int. J. Geograph. Inf. Sci.*, vol. 31, no. 11, pp. 2216–2233, Nov. 2017.

[26] T. Jia and Z. Ji, "Understanding the functionality of human activity hotspots from their scaling pattern using trajectory data," *ISPRS Int. J. Geo-Inf.*, vol. 6, no. 11, p. 341, Nov. 2017.

[27] W. R. Tobler, "A computer movie simulating urban growth in the detroit region," *Econ. Geograph.*, vol. 46, pp. 234–240, Jun. 1970.

[28] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[29] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2015, pp. 802–810.

[30] G. Huang, Z. Liu, L. Maaten, and K. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4700–4708.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, vol. 37. Lille, France, Jul. 2015, pp. 448–456.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015, pp. 1–15.

Tao Jia received the Ph.D. degree from the KTH Royal Institute of Technology, Sweden, in 2012. He is currently an Associate Professor with the School of Remote Sensing and Information Engineering, Wuhan University, China. His research interests include geographical information science in general, spatiotemporal data mining, complex network analysis and modeling, traffic analysis, and human dynamics in particular.

Penggao Yan received the B.Eng. degree from Wuhan University, China, in 2018, where he is currently pursuing the master's degree with the School of Remote Sensing and Information Engineering. His research interests include machine learning, traffic planning, and smart cities.

View publication stats
Authorized licensed use limited to: Wuhan University. Downloaded on September 11,2020 at 08:10:19 UTC from IEEE Xplore. Restrictions apply.